

ROB316 - PLANIFICATION ET CONTRÔLE

TP n°1 Contrôle de Robots Unicycle et Bicyclette

9 décembre 2019

Gabriel Henrique Riqueti

Victor Kenichi Nascimento Kobayashi

ENSTA IP Paris

Résumé

Ce cours, d'une manière générale s'appuie sur les modèles cinématiques (unicycle, tricycle, voiture et chariot avec remorque) bien comme ces applications au contrôle comme la stabilisation de trajectoires, le suivi de chemin et la stabilisation de configurations. Finalement, il y a une petite introduction concernant le régulateur PID, ses paramètres, avantages et Inconvénients.

Description du TP

Le TP1 nous introduit deux modèles de contrôle, le modèle unicycle et bicyclette bien comme les différents types de situation d'une suivie de chemin comme vers un point ou vers une position. L'objet du TP est de nous familiariser avec ces méthodes en les misant en oeuvre et comprendre l'importance du règlement de gains du contrôleur pour atteindre la condition donnée.

Contrôle de Modèle Unicycle

Question 1 - Contrôle Unicycle

En implémentant la méthode proposée dans cette question et en tournant le fichier `UnicycleToPose.m`, on voit que le modèle de robot unicycle commence à une position et une pose aléatoires et souvent le point de destination n'est pas devant le robot. Comme c'était déjà prévu, le robot tourne vers le point de destination sur place et, ensuite, il va vers le point destiné. Quand il est proche du point désiré, la position du centre du robot est déjà presque correcte, donc la vitesse est faible et c'est plus facile de régler la position angulaire. Ce parcours peut être vu dans la Figure 4.

Cette commande est réglée par quatre paramètres : K_{ϕ} , K_{α} , K_{β} et α_{\max} , qui

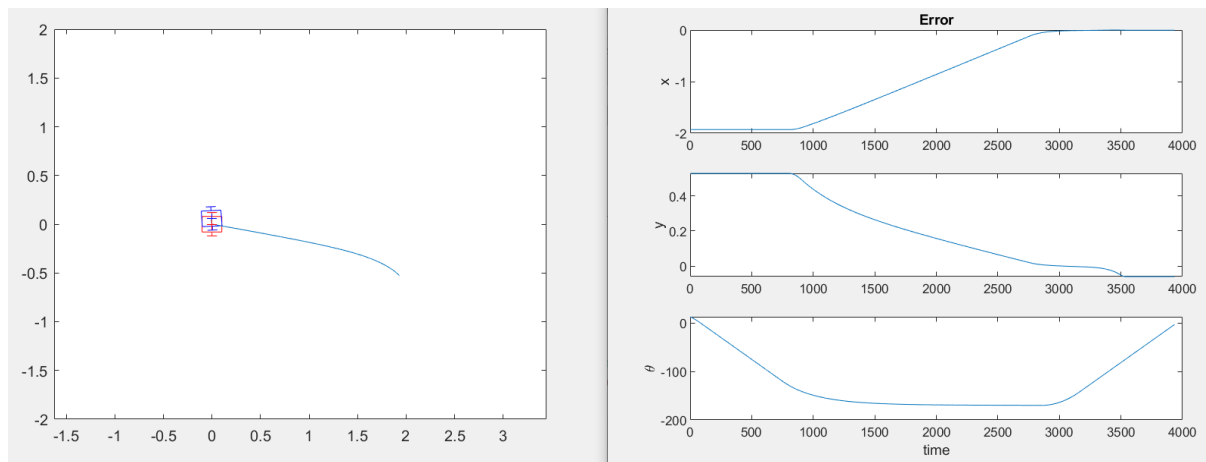


FIGURE 1: Chemin suivi par le modèle de robot unicycle vers une pose désirée (à droite) et l'erreur de chaque composant de position du robot (à gauche).

sont les gains proportionnels de distance, angulaire vers le point désiré et angulaire entre la pose désirée et actuelle et la valeur absolue de alpha tel que le robot peut marcher encore, respectivement. Le meilleur temps obtenu depuis la fonction `UnicycleToPoseBenchmark` était 2429,67 ms avec $K_{pho}=9$, $K_{alpha}=4$, $K_{beta}=11$ et $alpha_max=\pi/5$.

Contrôle de Modèle Bicyclette

Question 2 - Contrôle de Bicyclette vers un Point

En suivant l'algorithme donnée dans cette question, on a écrit le code `BicycleToPoint` qui contrôle le modèle bicyclette du robot vers une position sans prendre en compte l'orientation finale. Le comportement du robot avec ce contrôle est montré dans la Figure 2.

Concernant la performance, cette méthode a atteint jusqu'à 1390,2381 ms avec les paramètres $K_{rho}=18$ et $K_{alpha}=10$. On aperçoit que le robot peut aller plus vite au destin parce qu'il ne fait pas attention à l'orientation de la position finale.

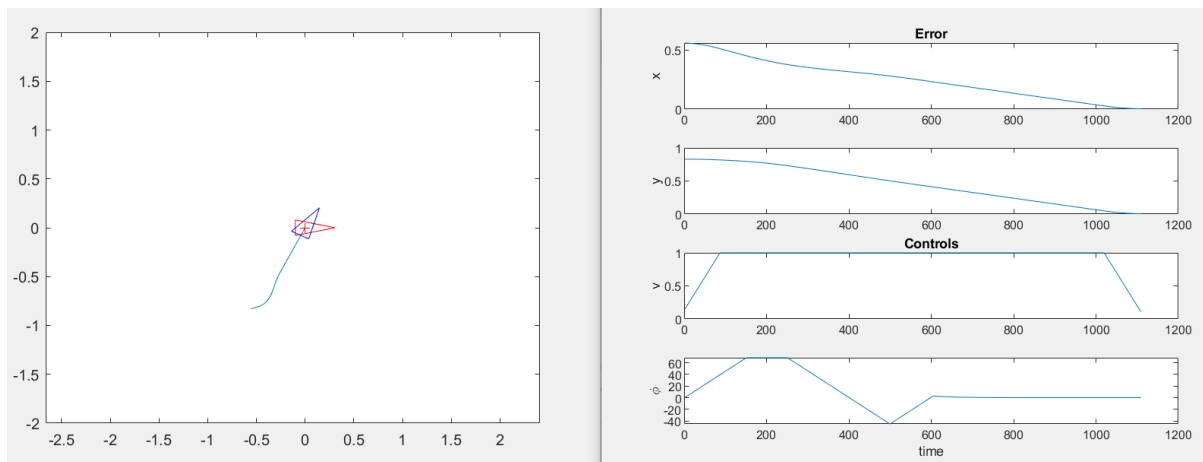


FIGURE 2: Chemin suivi par le modèle de robot bicyclette vers un point désiré (à droite) et l'erreur et les variables de contrôle vitesses linéaire et angulaire du robot (à gauche).

Question 3 - Contrôle de Bicyclette vers une Position

En écrivant le contrôleur proportionnel décrit par les équations de la question 3 nous obtenons un contrôleur qui guide le robot vers une orientation finale fixée. Cette trajectoire suivie et les erreurs de position sont représentés par la Figure 3 à gauche et à droite, respectivement.

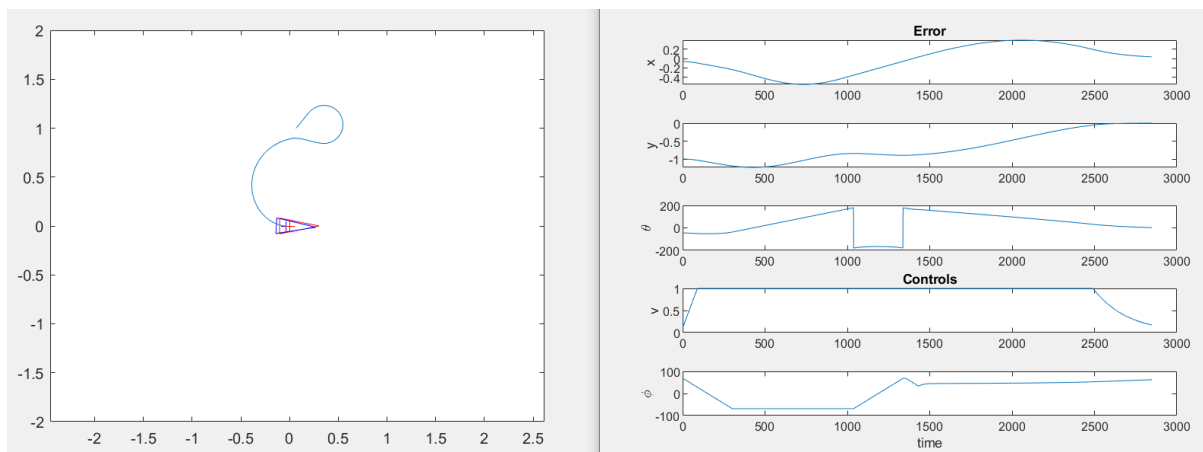


FIGURE 3: Chemin suivi par le modèle de robot bicyclette vers une pose désirée (à droite) et l'erreur et les variables de contrôle vitesses linéaire et angulaire du robot (à gauche).

Par ailleurs, en ce qui concerne, la régulation des gains, cette partie a été faite d'une façon empirique, c'est-à-dire, non seulement en regardant l'erreur de positionnement finale par rapport la position fixée mais aussi le score obtenu par le fichier BicycleToPoseBenchmark

qui a eu 1807.1429 ms comme meilleur temps d'exécution.

La régulation finale de gains a été faite comme sur la liste ci-dessous :

- $K_{rho} = 5$
- $K_{alpha} = 20$
- $K_{beta} = -9$

Question 4 -Contrôle de Bicyclette selon un Chemin

Pour cette question, on a utilisé la méthode proposée dans cette question pour faire le modèle bicyclette suivre un chemin déjà défini. En plus, on a crée une liste de points également séparés (par la distance step) entre les points consécutifs du chemin prédéfini afin de que l'algorithme toujours interprète le point devant et suivant du chemin comme le point à suivre. Les résultats sont affichés dans la Figure 4.

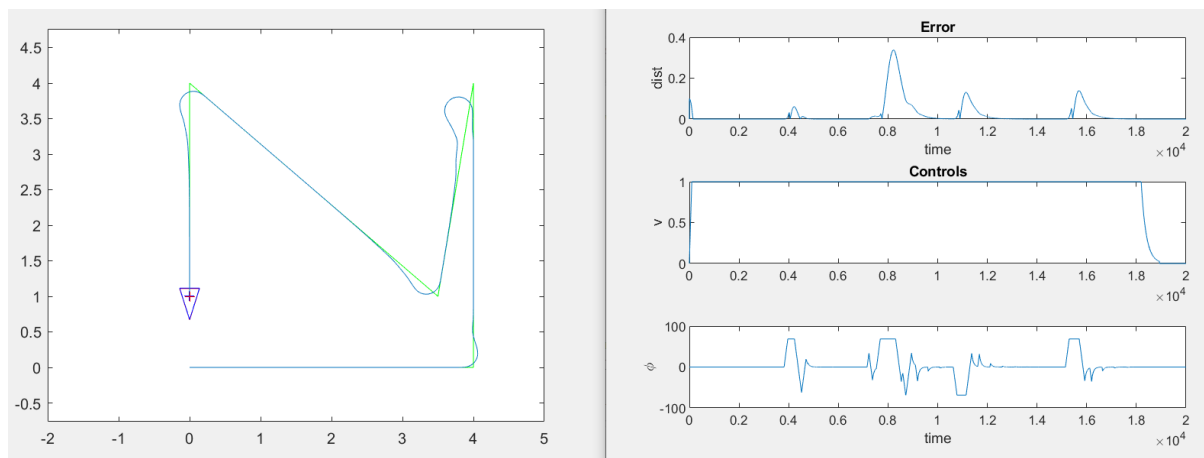


FIGURE 4: Chemin suivi par le modèle de robot bicyclette (en bleue) par rapport le chemin désiré (en vert) (à droite) et l'erreur et les variables de contrôle vitesses linéaire et angulaire du robot (à gauche).

En tournant le fichier BicycleToPath avec des paramètres $K_{rho}=5$, $K_{alpha}=7,8$ et $step=0.49$, le robot réussit à suivre le chemin avec l'erreur totale de 433,2748. Comme tous les autres questions, on remarque que l'augmentation des gains proportionnels angulaires engendre plus d'oscillation à partir d'une valeur.